

(11)Publication number : 05-225149

05-225149

(43) Date of publication of application : 03.09.1993

(51)Int.Cl.

G06F 15/16

G06F 9/46

G06F 12/00

(21)Application number : 04-025619

(71)Applicant : TOSHIBA CORP

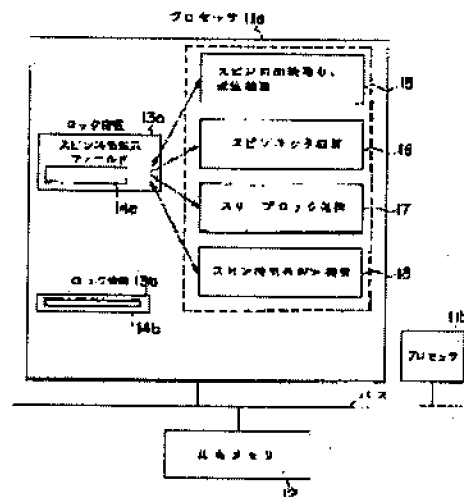
(22)Date of filing : 13.02.1992

(72)Inventor : KITSU TOSHIKI

(57)Abstract:

PURPOSE: To improve the efficiency of process execution by reducing the wasting of a processor at the time of lock acquisition.

CONSTITUTION: When the lock of a lock variable 13a can not be acquired, the process is set in a spin lock state by a spin lock mechanism 16 and when the lock can not be acquired even after a specific spin lock time spin, a sleep lock mechanism 17 sets the process in a sleep state. Consequently, the process is put in the sleep state unless the lock is obtained for a time exceeding the specific spin lock time set in a spin time indication field, and the processor is passed to another process; when the lock is released from another processor relatively early, the lock is acquired since the process is in the spin lock state, so that the process can be executed.



[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁(J P)

(12)公開特許公報(A)

(11)特許出願公開番号

特開平5-225149

(43)公開日 平成5年(1993)9月3日

(51)Int.Cl. ⁵	識別記号	片内整理番号	F I	技術表示箇所
G 0 6 F 15/16	3 5 0 F	8840-5L		
9/46	3 6 0 B	8120-5B		
12/00	5 7 2	8841-5B		

審査請求 未請求 請求項の数3(全 7 頁)

(21)出願番号 特願平4-25619

(22)出願日 平成4年(1992)2月13日

(71)出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72)発明者 岐津 俊樹

神奈川県川崎市幸区小向東芝町1番地 株
式会社東芝総合研究所内

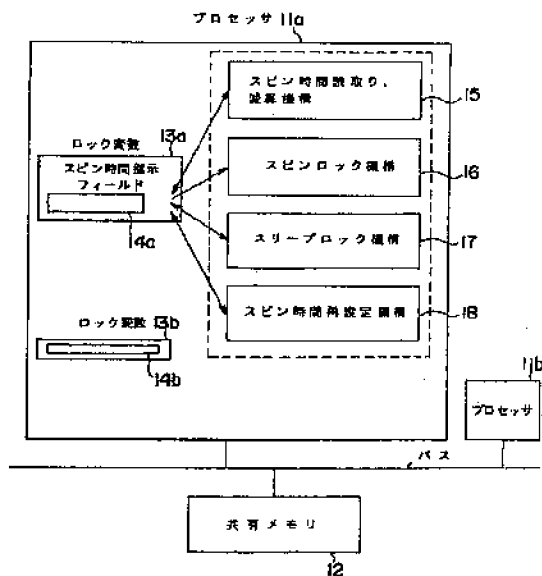
(74)代理人 弁理士 鈴江 武彦

(54)【発明の名称】 ロック方式

(57)【要約】

【目的】ロック獲得の際のプロセッサの浪費を少なくし、プロセス実行の効率化の向上を図る。

【構成】ロック変数13aのロックを獲得できない場合には、そのプロセスはスピンロック機構16によってスピンロック状態に設定され、そして所定のスピンロック時間スピンしてもロックを獲得できない場合にスリープロック機構17によってスリープ状態に設定される。このため、スピン時間指示フィールドに設定された所定のスピンロック時間を越える長い時間ロックが取れない時にはスリープとなって、プロセッサを他のプロセスに渡すことができ、また、ロックが他のプロセスから比較的早く解放された場合にはスピンロック状態にあるので直ぐにそのロックを獲得して実行することができる。



1

2

【 特許請求の範囲】

【 請求項1 】 共有変数を用いてプロセス間の排他制御を実現するメモリ共有型マルチプロセッサシステムにおいて、

前記各共有変数毎にロック待ちプロセスがスピンするスピ
ンロック時間を規定し、

ロックを獲得できなかったロック待ちプロセスを前記規定されたスピンロック時間だけスピンさせ、そのスピン
ロック時間内にロックを獲得できないときにそのプロセス
をスリープさせることを特徴とするマルチプロセッサ
システムのロック方式。 10

【 請求項2 】 前記スピンロック時間は、プロセスのス
リープおよびそのスリープ状態からの復帰のために要する
プロセッサの実行時間と実質的に等しい値に初期設定
されていることを特徴とする請求項1記載のマルチプロ
セッサシステムのロック方式。

【 請求項3 】 共有変数を用いてプロセス間の排他制御
を実現するメモリ共有型マルチプロセッサシステムにおいて、

前記各共有変数毎にロック待ちプロセスがスピンするス
ピンロック時間を割り当てる手段と、

ロックを獲得できなかったプロセスを前記スピンロック
時間だけスピンさせる手段と、

ロック解除待ちプロセスを有する共有変数に対応した前
記スピンロック時間の値を減少する手段と、

前記スピンロック時間内にロックを獲得できないときに
そのプロセスをスリープさせる手段と、

ロック獲得後そのロックを解放する際に前記スピン時間
を再設定する手段とを具備したことを特徴とするマルチ
プロセッサシステムのロック方式。 30

【 発明の詳細な説明】

【 0001 】

【 産業上の利用分野】 この発明は高性能な並列処理を実
現するための共有メモリを有したマルチプロセッサシ
ステムの排他制御におけるロック方式に関するものであ
る。

【 0002 】

【 従来の技術】 近年、共有メモリを有したマルチプロセ
ッサシステムにおいては、排他制御のための種々のロッ
ク方式が提案されている。ロック方式には、大別してプ
ロセスがロックを取れるまで共有変数(ロック変数)に
アクセスし続けるスピンロック方式と、ロックが取れ
なかった場合スリープし、ロックが解放された時に起こ
されてからロックを取りに行くスリープロック(サスペ
ンドロックと称する場合もある)方式とがある。 40

【 0003 】 スピンロック方式では、スピンしている期
間はプロセッサは浪費されることになる。一方、スリー
プロック方式では、プロセスのスリープとそのスリー
プからの復帰(ウェイクアップ)の際にレジスタ退避やレ
ジスタ復帰が必要になるため、プロセッサ(CPU)実 50

行時間が余分にかかるその分だけコストが大きくなる
が、プロセスがスリープしている間はプロセッサは他の
プロセスの実行に使用でき、プロセッサが浪費されるこ
とはない。

【 0004 】 従来のロック方式では、資源がロックされ
ている時間がわからないため、予めスピンするかスリー
プするかが決められており、長い時間ロックが取れない
時にスピンしたり、ロックがすぐ取れる時にスリープす
るなど、プロセッサを浪費してしまうという問題があっ
た。

【 0005 】

【 発明が解決しようとする課題】 従来のロック方式にお
いては、ロックを取得する際に予めスピンするかスリー
プするかを決定しておくが、資源がロックされる時間が
わからないために無駄にスピンしたりスリープしてしま
うという問題があった。

【 0006 】 この発明はこのような点に鑑みてなされた
ものであり、ロック獲得の際のプロセッサの浪費を少な
くし、効率の良いプロセスの実行の実現することができ
るマルチプロセッサシステムのロック方式を提供する
ことを目的とする。

【 0007 】

【 課題を解決するための手段および作用】 この発明のロ
ック方式は、共有変数を用いてプロセス間の排他制御を
実現するメモリ共有型マルチプロセッサシステムにおい
て、前記各共有変数毎にロック待ちプロセスがスピンす
るスピンロック時間を規定し、ロックを獲得できなかった
ロック待ちプロセスを前記規定されたスピンロック時
間だけスピンさせ、そのスピンロック時間内にロックを
獲得できないときにそのプロセスをスリープさせること
を特徴とする。

【 0008 】 このロック方式においては、ロックを獲得
できない場合には、そのプロセスはまずスピンロック状
態に設定され、そして、所定のスピンロック時間スピン
してもロックを獲得できない場合にスリープ状態に設定
される。このため、所定のスピンロック時間を越える長
い時間ロックが取れない時にはスリープとなって、プロ
セッサを他のプロセスに渡すことができ、また、ロック
が他のプロセスから比較的早く解放された場合にはスピ
ンロック状態にあるのですぐにそのロックを獲得して実
行することができる。したがって、不必要に長い時間ス
ピンしたり、短い時間スピンすればロックが獲得でき
るのにスリープするといったプロセッサの浪費が避けら
れ、プロセスの効率的な実行が可能となる。

【 0009 】

【 実施例】 以下、図面を参照してこの発明の実施例を説
明する。

【 0010 】 図1にはこの発明の一実施例に係わるマル
チプロセッサシステムのシステム構成が示されている。
このマルチプロセッサシステムは、メモリ共有型のマル

3

チプロセッサであり、共有変数を用いて排他制御を実現している。

【0011】このマルチプロセッサシステムは、複数のプロセッサ11a, 11b, …と、共有メモリ12とから構成されている。複数のプロセッサ11a, 11b上で並列動作される複数のプロセスは、共有メモリ12上の共有データを同時に操作しないように、その共有データをロックして排他的に使用する。この排他制御のための同期は、共有変数(ロック変数)を用いて実現されている。

【0012】ロック変数はあるリソースの使用状態を示すものであり、例えば、ロック変数の値が1の時は、ある1つのプロセスが排他的な処理を実行中であること(ロック状態)を示し、他のプロセスはその処理を実行することができない。

【0013】ロック確保時のロック変数の参照、更新は、例えば、テストアンドセット命令(test & set)のように参照と変更が不可分に行われるプロセッサ命令で行なわれる。テストアンドセット命令は、基本的には不可分の単一機会命令として、指定されたロック変数から値“0”を読み出した後、“1”の値をロック変数に書き戻すものであり、最初に“0”(ロックがオフ)を受け取るプロセスだけがロックを確保して処理を実行でき、それ以外のプロセスはすべて“1”(ロックがオン)を受け取ることにより実行を待たされる。

【0014】この場合、スピンロック方式においては、“1”を受け取った各プロセスは、ロック変数が“0”になるまで繰り返しテストを行い、ロック解除されまで待つ。この状態は、スピングループと称されている。一方、スリーブロック方式においては、“1”を受け取った各プロセスは、プロセッサを解放するスリープ状態となり、ロック解除の合図を待ってウェイクアップされる。

【0015】ロック変数13a, 13b…は、共有メモリ12上の共有データの排他制御のためにそれら共有データ毎に生成されるものであって、各ロック変数13a, 13b, …にはスピン時間指示フィールド14a, 14b, …が割り当てられている。

【0016】各スピン時間指示フィールド14a, 14bには、対応するロック変数によってロック待ちとなったプロセスに対してスピンすべき時間を指示するためのスピン時間が設定される。このスピン時間の初期値は、スピンし続けるコストと、スリープし、ウェイクアップするコストとが等しくなるような値に設定される。

【0017】この場合、スリープし、ウェイクアップするコストとは、プロセスのスリープおよびウェイクアップのために必要なレジスタ退避および復帰に要するプロセッサ実行時間(実行サイクル数)を示す。このため、スピン時間は、プロセスのスリープおよびウェイクアップのために必要なプロセッサ実行時間と実質的に同じ値

4

に予め設定される。これにより、スピンしている間にロックが獲得できた場合には、スリープするよりも低コストになることが保証される。

【0018】また、各プロセッサ11A, 11b, …には、ロック変数13a, 13b, …を用いたロック方式を実現するための機構として、スピン時間読取り・減算機構15、スピンロック機構16、スリーブロック機構17、スピン時間再設定機構18が設けられている。

【0019】次に、図2のフローチャートを参照して、これらスピン時間読取り・減算機構15、スピンロック機構16、スリーブロック機構17、およびスピン時間再設定機構18の機能、およびこれらを利用したロック動作を説明する。

【0020】プロセスが資源を確保する際には、まず、該当するロック変数13aおよびそのロック変数に対応するスピン時間フィールド14aが参照され、スピン時間読取り・減算機構15によって上記フィールドの値を記憶した上で(ステップa1)そのフィールドの値が減少される(ステップa2)。

【0021】減少される値は任意の値で良く、ここではd1とする。これは次にロックを取りに来るプロセスはロックを取りにくくなっており、スピンする時間を短くすることによってスピンによるプロセッサの浪費を減少させるためである。

【0022】次に、ロック変数の値が“0”か“1”かによってロックを確保できるか否かを判断し(ステップa3)、ロック変数の値が“1”の場合には、そのプロセスはロックを確保できない。このロックを確保できなかったプロセスはスピンロック機構16によってスピンされ、ステップa1で読み取ったスピン時間だけスピンしてその間、繰り返しロックを獲得しに行く(ステップa3, a4)。

【0023】スピン時間内にロックが獲得できなかった場合には、さらに上記スピン時間指定フィールドの値がスリーブロック機構17によって減少され(ステップa5)、そして、ロックが獲得できなかったプロセスがスリーブロック機構17によってスリープされる(ステップa6)。この時に減少させる値も任意の値で良く、ここではd2とする。

【0024】この場合に、上記フィールドの値を再び減らすのは、スピン時間中にロックが取れないほどロックが獲得しにくくなっているため、さらにスピンする時間を短くすることによってスピンによる他のプロセッサの浪費を減少させるためである。

【0025】スリープされたプロセスはロックが解放されたときに通知等によってウェイクアップされ(ステップa7)、ロックを確保しに行く。ここでロックを確保できた場合には、そのプロセスによって所定の作業が実行される(ステップa9)。この後、スピン時間再設定機構18によって、ロックを獲得するまでに減じた値分

10

20

30

40

50

5

だけ上記フィールドの値が増加される(ステップa 10)。

【0026】この例では、スピンしている間にロックが獲得できた場合には、d 1 を、スピンしている間にロックが獲得できずスリープした後に獲得した場合には、d 1 + d 2 を、上記フィールドの値に加算して初期値に戻す。

【0027】このように上記フィールドの値を増加させるのは、ロックが解放されたことによりロックが解放される前に比べ獲得しやすくなるため、スピンする時間を長くしてスピンによりロックを獲得しやすくするためである。その後、プロセスは、ロックを解放する(ステップa 11)。次いで、図3のフローチャートを参照して、図1のシステムにおける別のロック動作を説明する。

【0028】プロセスが資源を確保する際には、まず、該当するロック変数13 a およびそのロック変数に対応するスピン時間フィールド14 a が参照され、スピン時間読取り・減算機構15によって上記フィールドの値を記憶した上で(ステップb 1) そのフィールドの値が減少される(ステップb 2)。

【0029】減少される値は任意の値で良く、ここではd 1 とする。これは次にロックを取りに来るプロセスはロックを取りにくくなっており、スピンする時間を短くすることによってスピンによるプロセッサの浪費を減少させるためである。

【0030】また、同時に時刻を記録するフィールドに現在の時刻を、スピン時間読取り・減算機構15によって記録しておく(ステップb 3)。次に、ロック変数の値が“0”か“1”かによってロックを確保できるかを判断し(ステップb 4)、ロック変数の値が“1”の場合には、そのプロセスはロックを確保できず、スピンロック機構16によってスピンされ、ステップb 1 で読み取ったスピン時間だけスピンしてその間、繰り返しロックを獲得しに行く(ステップb 4、b 5)。

$$x' = \{(n+1)/n\}x - (y/n) + d \quad \dots(1)$$

で与えられる。しかし、この(1)式で求めた x' は、※とし、(1)式で求めた x' が $x' \geq x_0$ の場合には、求めた x' が $0 < x' < x_0$ の範囲にある場合のみ有効※

$$x' = x_0 \quad \dots(2)$$

とし、また、(1)式で求めた x' が $x' \leq 0$ の場合には、

$$x' = 0 \quad \dots(3)$$

とすることが好ましい。

【0036】これにより、スピンする時間は上記のnの大小によって振幅が変化するが、徐々にその資源を獲得するまでの時間の平均値に収束していく。これにより資源を獲得する際には、実際にロック獲得までに要した平均的な時間分だけスピンするようになり、プロセッサの浪費が減少することになる。

【0037】以上のように、この実施例のロック方式においては、ロックを獲得できない場合には、そのプロセ

6.

*【0031】スピン時間内にロックが獲得できなかった場合には、さらに上記スピン時間指定フィールドの値がスリープロック機構17によって減少され(ステップb 6)、そして、ロックが獲得できなかったプロセスがスリープロック機構17によってスリープされる(ステップb 7)。この時に減少させる値も任意の値で良く、ここではd 2 とする。

【0032】この場合に、上記フィールドの値を再び減らすのは、スピン時間中ではロックが取れないほどロックが獲得しにくくなっているため、さらにスピンする時間を短くすることによってスピンによる他のプロセッサの浪費を減少させるためである。

【0033】スリープされたプロセスはロックが解放されたときに通知等によってウェイクアップされ(ステップb 8)、ロックを確保しに行く。ここでロックを確保できた場合には(ステップb 9)、スピン時間再設定機構18によって、ロックをとれた時刻とスピンを開始した時刻との差により、ロック獲得に要した時間が求められる(ステップb 10)。また、そのロックを獲得したプロセスによって所定の作業が実行される(ステップb 11)。

【0034】この後、スピン時間再設定機構18によって、ロックを獲得するまでに減じた値(d 1、またはd 1 + d 2)と、ロック獲得に要した時間と、ステップb 1 で記憶したもとのスピン時間とから、新たなスピン時間が求められ、それがスピン時間指示フィールドに再設定される(ステップb 12)。そして、その後、ロックが解放される(ステップb 13)。この再設定される新たなスピン時間 x' は、例えば、次のように求められる。

【0035】 x' を新しいスピン時間、 x を読み取ったスピン時間、 x_0 をスピン時間の初期値、 y をロックを獲得するまでにかかった時間、 d をロックを獲得するまでに減算した値、 n を任意の正数とすると、

スとはまずスピンロック状態に設定され、そして、所定のスピンロック時間スピンしてもロックを獲得できない場合にスリープ状態に設定される。このため、所定のスピンロック時間を越える長い時間ロックが取れない時にはスリープとなって、プロセッサを他のプロセスに渡すことができ、また、ロックが他のプロセスから比較的早く解放された場合にはスピンロック状態にあるのですぐにそのロックを獲得して実行することができる。

【0038】したがって、不必要に長い時間スピンした

り、短い時間スピンすればロックが獲得できるのにスリープするといったプロセッサの浪費が避けられ、プロセスの効率的な実行が可能となる。

【0039】また、スピンする時間を静的、あるいは動的に変更することにより、ロックされている時間の変動に柔軟に対処でき、効率の良いプロセスの実行が可能となるなど多大なる効果が得られる。

【0040】

【発明の効果】以上説明したようにこの発明によれば、ロック獲得の際のプロセッサの浪費を少なくでき、効率の良いプロセスの実行の実現することが可能となる。

【図面の簡単な説明】

【図1】この発明の一実施例に係わるシステム構成を示すブロック図。

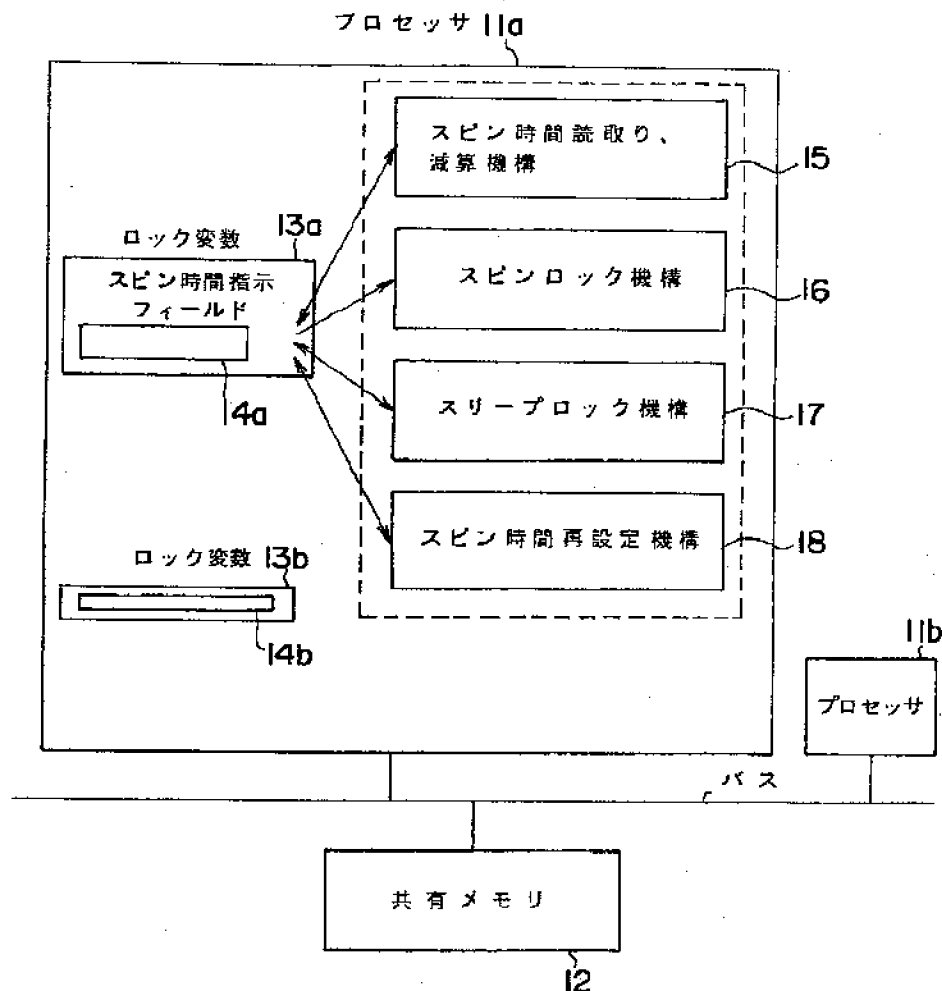
【図2】同実施例のロック動作の一例を説明するフローチャート。

【図3】同実施例のロック動作の他の例を説明するフローチャート。

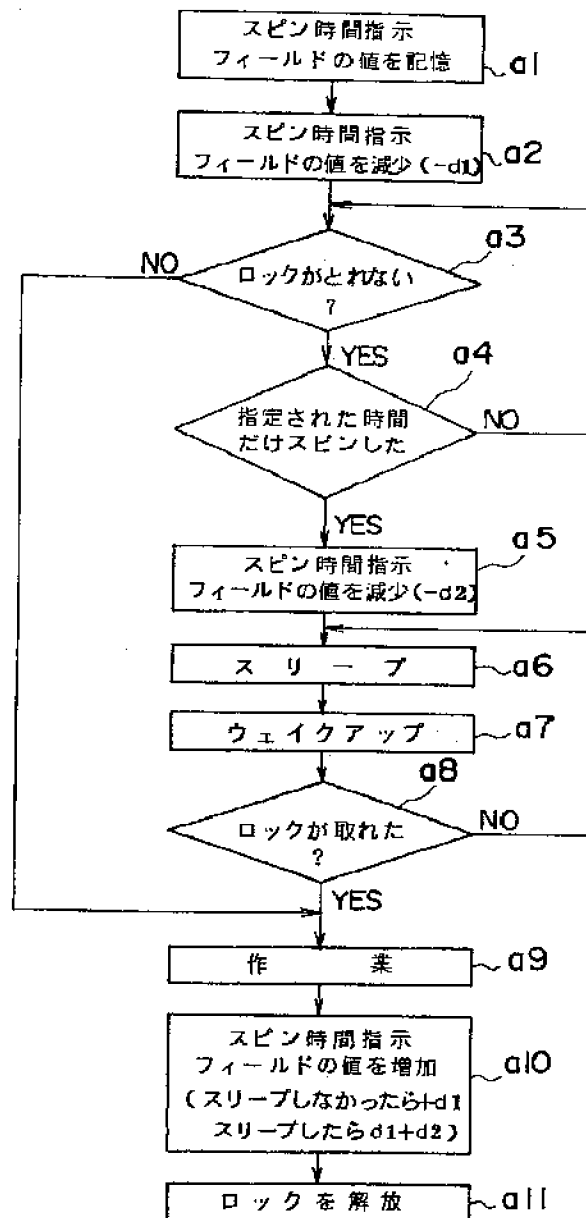
【符号の説明】

11a, 11b…プロセッサ、12…共有メモリ、13a, 13b…ロック変数、14a, 14b…スピン時間指示フィールド、15…スピン時間読取り・減算機構、16…スピンロック機構、17…スリープロック機構、18…スピン時間再設定機構。

【図1】



【 図2 】



【 図3 】

